

# **DZERO Level 1 Central Track Trigger (CTT)**

## **Level 1 - Central Tracker Octant Card (L1CTOC) Firmware Design**

**Version: Draft 0.1**

**Date: 12 September 2002**

**Ricardo Ramirez**

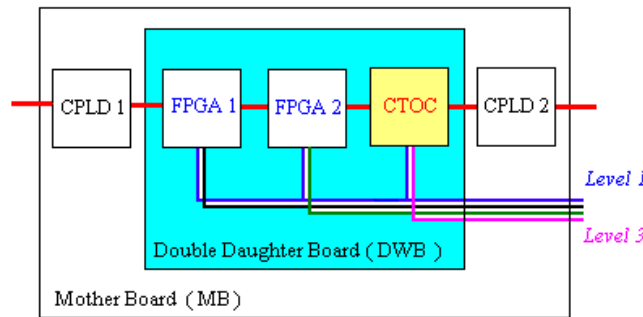
## Table of Contents

Table of Contents .....	1
1. Introduction .....	2
1.1 Overview .....	2
2. Protocols .....	3
3. Overview of L1 CTOC .....	4
3.1 Coding structure .....	4
3.1.1 Finite State Machine .....	4
3.1.2 Basic Module .....	4
3.1.3 If ... Case .....	5
3.2 General Functional Structure .....	5
3.2.1 Receiver .....	5
3.2.2 Body of Calculations .....	6
3.2.3 Level 1 Sender .....	10
4. L1 CTOC Package Location and Organization .....	11
4.1 Packages .....	11
4.2 Folders .....	11
5. L1 CTOC VHDL Components .....	12
5.1 LIFE .....	12
5.2 Body of calculations .....	12
5.2.1 L1CTOC.vhd .....	12
5.2.2 Pipelined_Add.vhd .....	13
5.2.3 Pipeline_BiggetSumPt.vhd .....	14
5.2.4 L1CTOC_EoR.vhd .....	14
5.2.5 Or_empty.vhd .....	14
5.2.6 Pt_Sum.vhd .....	14
5.2.7 Tracks_sum.vhd .....	15
5.2.8 Occ_Lev.vhd .....	15
5.2.9 PS.vhd .....	15
5.2.10 Biggest_SumPt.vhd .....	15
5.2.11 Iso_Tracks.vhd .....	16
5.2.12 GI.vhd .....	16
5.2.13 Header_MV.vhd .....	16
5.2.14 Revision_reg.vhd .....	16
5.2.15 RegPresender.vhd .....	17
5.3 L1 Sender .....	17
6. Configuration Parameters .....	17
7. Scripts .....	17
8. Timing Simulations .....	18
8.1 Latency .....	18
8.2 Body of calculations latency .....	18
9. 1553 interaction .....	18
10. Implementation issues .....	19
11. Outlook .....	19
12. Glossary .....	20
13. References .....	21

# 1. Introduction

This firmware has been placed in a Digital Front End Boards (DFE) [3], type Double Wide Daughter Board (DWDB).

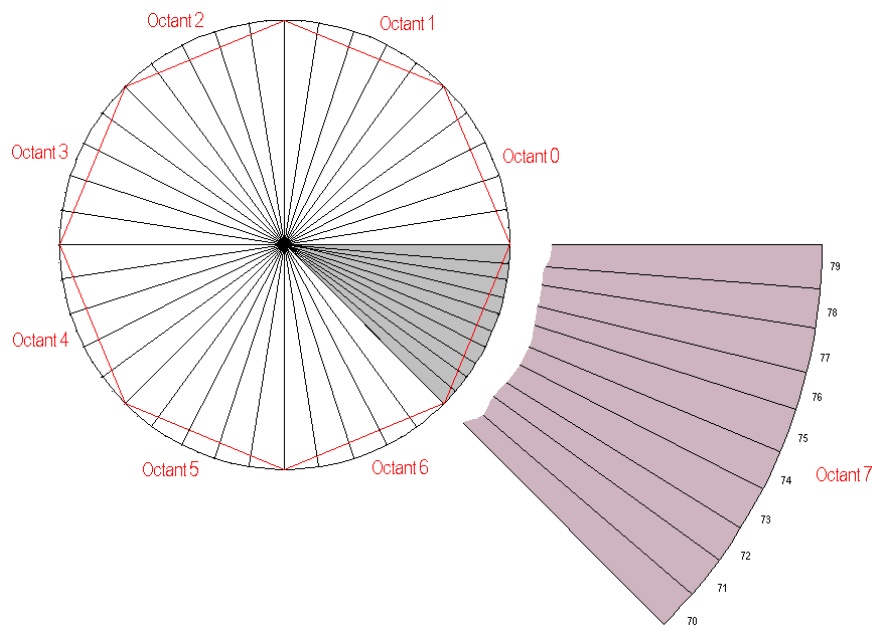
This DFE, denominated Central Tracker Octant Card (CTOC) has three Xilinx 600BG560 FPGAs [2], which have a BGA (Ball Grid Array) [4] disposition for their pins. L1CTOC is the third device in the Double Wide Daughter Board (DWDB), and it is the last one to be read in the JTAG chain for DWDB. Also, it shares the Level 3 information bus with the others two FPGAs disposed in the DWDB. The JTAG configuration line goes from CPLD 1 to CPLD 2, so L1CTOC is the fourth device in the chain for the MB, but the third one for DWDB



**Figure 1** CTOC Board Disposition.

## 1.1 Overview

The L1CTOC firmware [1] is the piece of code in charged of processing the information coming from the 10 Digital Front End Analog Board (DFEA) outputs, which process each Octant of the Central Fibber Tracker (CFT).



**Figure 2** CFT Octant distribution

## 2 Protocols

According to the latest version of the protocols [5], L1 CTOC has 10 LVDS inputs which carry 196 bits arranged in 7 meaningful frames of 28 bits each one.

Frame	BoR			Hp	First Byte								Second Byte								Third Byte								
	27	26	25		24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1	1	1	1	Hp	P&M		OCT		SCL		0	0	0	0	0	RA TS CFT.CPSa			L1 CTT/PS Data				L1/L2						
	196	195	194		193	192	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176	175	174	173	172	171	170	169
F2	0	0	0	Hp	XPT+LC			XPT-LC			0	0	XPT+TC			XPT-TC			0	0	XPT+NC		XPT-NC		0	0	142	141	
	168	167	166		165	164	163	162	161	160	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144	143	142	141
F3	0	0	0		HPT+LC			HPT-LC			0	0	HPT+TC			HPT-TC			0	0	HPT+NC		HPT-NC		0	0	113		
	140	139	138		137	136	135	134	133	132	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
F4	0	0	0	Hp	MPT+LC			MPT-LC			0	0	MPT+TC			MPT-TC			0	0	MPT+NC		MPT-NC		#PSC				
	112	111	110		109	108	107	106	105	104	103	102	101	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85
F5	0	0	0	Hp	LPT+LC			LPT-LC			0	0	LPT+TC			LPT-TC			0	0	LPT+NC		LPT-NC		0	0	57		
	84	83	82		81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57
F6	0	0	0	Hp	Sum Abs [Pt]						0	0	Occ Level TS						Isolated Track										
	56	55	54		53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
F7 / Tp	0	0	0		Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	
	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 3 Level 1. Input Protocol. DFEA - L1CTOC

Frame	BoR			Hp	First Byte												Second Byte								Third Byte							
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
F1 / HF	1	1	1	Hp	P&M		OCT			#OBJ VP			#OBJ MSB		#OBJ LSB		#OBJ HP		L1 CTT/PS Data Type				L1/L2									
F2 / HF	0	0	0	Hp	Turn Number																Crossing Number											
F3 / D1	0	0	0	Hp	RA PSC CPSax				RA TS CFT/CPSax				T	L	C	Pt. Bin		Pt. Ext.		R	RA CFT H-Doublet				ISO							
----	0	0	0	Hp	-----																											
Fn / TF	0	0	0	0	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp				

Figure 4 Level 2. Input Protocols. DFEA - CTOC

Frame	BoR			Hp	First Byte								Second Byte								Third Byte							
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1	1	1	1	Hp	XPT+LC		XPT-LC		XPT+TC		XPT-TC		XPT+NC		XPT-NC		P&M		GI	L1/L2								
	196	195	194	193	192	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176	175	174	173	172	171	170	169
F2	0	0	0	Hp	HPT+LC		HPT-LC		HPT+TC		HPT-TC		HPT+NC		HPT-NC		Occ Lev		OCT	SCL								
	168	167	166	165	164	163	162	161	160	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144	143	142	141
F3	0	0	0	Hp	MPT+LC		MPT-LC		MPT+TC		MPT-TC		MPT+NC		MPT-NC		0	T1	T2	#PSC								
	140	139	138	137	136	135	134	133	132	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
F4	0	0	0	Hp	LPT+LC		LPT-LC		LPT+TC		LPT-TC		LPT+NC		LPT-NC		Sum Abs [Pi]											
	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85
F5	0	0	0	Hp	Isolated Track								Isolated Track								Isolated Track							
	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57
F6	0	0	0	Hp	OCT		RA HTPS		Isolated Track								Isolated Track											
	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
F7 / TF	0	0	0	0	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	Vp	
	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Figure 5 L1 Output. CTOC-CTTT

## 3 Overview of L1 CTOC

### 3.1 Coding structure

The main idea is about guarantee an optimum timing consumption and a better fitting into device.

#### 3.1.1 Finite State Machine

According to prevent a lack of reset signal, a software reset has been implemented. This signal is generated using a simple FSM, which uses a configuration depending of the EoR signal that is propagated through each module, and is named *ready*. So it becomes a software asynchronous reset using ready signals through the modules.

The basic process has the organization as is showed in below figure.

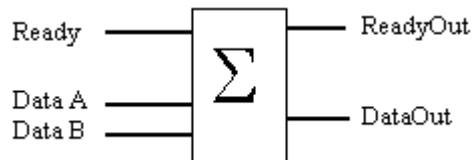
```
process (clock)
begin
    if (clock = '1' and clock'event) then
        if state = S0 then
            --
            -- conditions of reset or
            -- initial state
            --
            if ready = '1' then
                state <= S1;
            end if;
        elsif state = S1 then
            state <= S2;
            ...
            ...
            ...
        else
            state <= S0;
        end if;
    end if;
end process;
```

**Figure 6** Finite State Machine

#### 3.1.2 Basic Module

The previous version of CTOC offered a design with High Speed theoretical features. However, the practical way shows that this design cannot be implemented because of limitations in resources. The previous version had taken a lot of multiplexers and registers. However, it could be fixed into the 600 Virtex, but the timing requirements were not filled.

The new design offers pipelined structure based in a simple module, which makes simple operation such as additions and truncations. It can establish a high-speed featured design using parallelism and interacting with a few inputs and outputs.



**Figure 7** Basic Modules

This module uses a simple ready signal like a synchronization key. This signal comes from the L1FE module [6], and it is named EoR (End of Record). Each module divides its functions using few input and output signals, for the Xilinx Look Up Tables (LUTs) and makes operations with four inputs of one bit and obtains an answer that has only one bit.

This kind of distribution provides an excellent use of the logical resources.

In the CTOC simplest case, it requires at least two outputs. So the most simple unit will take at least two LUTs.

### 3.1.3 If ... Case

The considerations according to the type of coding required [8] by the Finite State machines was a important point considered in this design.

Using 'Case' statements forces to look for a better fitting into the devices. Although, it cannot produce all of the LUTs required for it. This problem could be fixed if a different compilation tool is used.

In the other hand, the 'if' statements definition takes a lot of logical device resources, but it can be implemented, again, it because our implementation tool. The tests have showed that this kind of coding is useful and practical.

## 3.2 General Functional Structure

The L1 CTOC has three major parts. The first one, named Receiver, takes the inputs signal and recognizes what kind of information is arriving. The next one, Body of Calculations, is in charge of make available all of the Track List and the required operations to combine CPS and CFT information. And the last one, Sender, which sends all of the information in Level 1 format.

### 3.2.1 Receiver

This module has been thought according to get all of the inputs coming from the LVDS links, which are 10 for each CTOC board.

This module discriminates if the input corresponds to Level 1 or Level 2, and finds the "Beginning of Record (BOR)". The BOR signs which is the very first frame. It is very useful in order to identify which is the start of each frame.

In the previous version a big OR was implemented, so the general BOR signal depended on the time in which the others links arrived. In the new version when a BOR from any link arrives, a signal is produced. This BOR waits until at least three or more BOR arrives, then send a BOR. For the old version this process was required in order to offer a  $\pm 1$  tick of de-skew. New tests have showed that this definition is inefficient and produces a lot of synchronization mistakes verifiable them in the output.

The same operation is realized for the Data Type and Pass Mark. In the new versions, it is made by Majority Votes modules which according to the number of links chooses for the most common value showed in the inputs.

## 3.2.2 Body of Calculations

It does the information required for the next level of trigger. This part is divided in small modules that take the information arriving in buses of data.

### 3.2.2.1 Pass and Mark

These two bits are coming in the Header frame (bits 23 and 22) of the CTOC input (DFEA output).

**Table 1** Pass and Mark

Value	Meaning
00	Normal
01	Trigger Framework (TFW) will pass this event on
10	
11	No used

### 3.2.2.2 Serial Command Link

This information lets CFT to do the synchronization with all of the other detector systems [\[9\]](#).

**Table 2** Serial Command Link

Header Bit number	Meaning
18	CFT Reset
17	First crossing

### 3.2.2.3 Occupancy Level

This module makes an addition of all of the inputs, and then it takes the result and makes a truncation. It can be explained because each of the links sends information about the Occupancy Level, which corresponds to its octant. In the CTOC case, the 10

percent of the final output is equivalent to 256 hits. In order to represent this number in binary, a word of 8 bits is required.

The truncation is made using only the 6<sup>th</sup> and 9<sup>th</sup> bits, which are taken as a trigger indicator.

**Table 3** Occupancy Level

Number of hits (Dec.)	Number of hits (Hex.)	Percentage of Total Occupancy [%]	Trigger to CTTT Board (Truncation)
480	111100000	20	11
375	101110111	15	10
250	011111010	10	01
< 127	001111111	< 5	00

### 3.2.2.4 Data Type

This frame is obtained using a simple OR, so the word that carries this information which is the First one or Header.

**Table 4** Data Type

Bits	Source
00101	L1 CFT/CPS Axial and L2CFT
00010	L1 FPS and L2 FPS
11000	L1 FPD and L2 CPS ax
11111	L2 STT

### 3.2.2.5 Octant ID

Each L1CTOC has its own Octant ID number, which can be between 0 and 7. It represented in three bits at the Header (bits 21 and 19).

### 3.2.2.6 Relative Address

This information is related to the Relative Address of the Octant with Highest Sum of Pt in a specific Trigger Sector. Four bits in the Header (bits 11 to 8) show this information.

### 3.2.2.7 Level 1 and Level 2 marker

This three bits mark the type of information flowing through L1CTOC. If Level 1 data is arriving the operations into the Body of Calculation are executed. Otherwise, when Level 2 information happens, the 'Level 3 Sender' starts sending a copy of the input of the L1CTOC input to Level 3. Also, in the Level 2 case, a copy of the 'DFEA Level 2 output', which carries information concerning about 'Turn Number' and 'Crossing Number', is sent to CTTT.

**Table 5** Level 1 / Level 2

Bits	Meaning
000	Level 2 information
001	
010	
100	
111	Level 1 Information
110	
101	
011	

### 3.2.2.8 Global Isolated

This module generates a single bit, which indicates that a least one isolated track in that Octant is observed with a determined occupancy value for that specific sector.

### 3.2.2.9 Isolated Tracks

The information coming from DFEA corresponds to each Trigger Sector, and L1CTOC puts in its output the information for each Octant.

This module compares pairs of frames of the information that arrives in each link, so it can choose the biggest isolated value that has arrived.

**Table 6** Bits - Isolated Tracks

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
V	Association			C	Pt		Occupancy

**Table 7** Isolated Track Frame

Information	Meaning	
V	'1' is an isolated track is found.	
Association	000	Not association found
	001	Tight assoc. with Low CPS
	010	Reserved
	101	Loose assoc. with Low CPS
	010	Reserved
C	Sign of Curvature, if '1' equals positive	
Pt	00	> 10 GeV
	01	5 – 10 GeV
	10	3 – 5 GeV
	11	1.5 – 3 GeV
Occupancy	'1' is the home trigger sector and its neighbours corresponds to a given value	

### 3.2.2.10 *Biggest Sum Pt*

This Biggest Sum Pt. makes choose between all of the links searching for the Biggest one, and then, showing its address (Relative Sector Address – RAS – and Octant ID numbers), plus its content.

### 3.2.2.11 *Sum Absolute Pt*

The information for Absolute Pt coming from DFEA is added and reported for each Octant (10 trigger sectors).

### 3.2.2.12 *Sum Pt*

This module makes the addition of the information, coming for each specific value of the Pt Bin input. These values are represented with two more features, the first one is the 'sign of curvature', and the second one is the association with the CPS cluster.

The association of the track with the CPS cluster is represented as follow.

**Table 8** CPS cluster association

Symbol	Meaning
NC	No associated
TC	Tightly associated
LC	Loosely associated

If the value of this addition is more than “110” (6 decimal) an overflow output is generated, putting “111” at the output of L1 CTOC for that specific value of Pt.

**Table 9** Pt Bin

Pt	Meaning		
00	XPT	Max	> 10 GeV
01	HPT	High	5 - 10 GeV
10	MPT	Medium	3 - 5 GeV
11	LPT	Low	1.5 - 3 GeV

**Table 10** Extended Pt

	Pt	Extended Pt	Average Pt
Max	00	000	80
	00	001	27
	00	010	16
	00	011	11
High	01	000	9
	01	001	7
	01	010	6
	01	011	5
Med	10	000	4
	10	001	4
	10	010	3
	10	011	3
Low	11	000-111	2

### 3.2.2.13 Number of Pre-shower Cluster

This module calculates the Number of Pre-shower Cluster located in that octant, so the information that arrives is by sector

### 3.2.2.14 CPS association

This function is not implemented yet. The terms T1 and T2 are turned on according to the corresponding CPS association.

**Table 11** CPS association

Bits	Meaning
T1	'1' if <b>one</b> CPS cluster is loosely associated
T2	'1' if at least two CPS cluster are loosely associated.

### 3.2.3 Level 1 Sender

This module is in charge of calculating the Vertical Parity and sending the information properly formatted to CTTT.

## 4 L1 CTOC Package Location and Organization

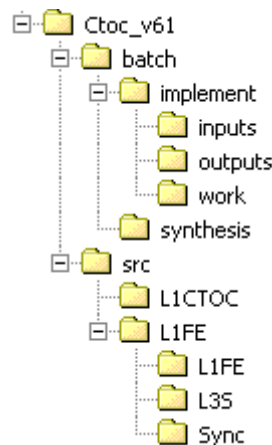
The basic organization for L1CTOC can

### 4.1 Packages

The VHD code requires three packages. They are, 'L1FEpack', 'L3pack', and 'CTOC\_pack'. The two firsts are used by L1FE in order to infer the basic features for the generic values in its code. 'CTOC\_pack' is used on principal for determining the value of the *OCT ID*, which is required for 'Biggest Sum Pt' module, and also it is sent to CTTT.

### 4.2 Folders

The basic required organization for implementation requests for implementation requests is as appears in the below figure. Note that the folders with the macros scripts (batch), and the source code (src) have a specific structure, which must be respected in order to use the macro resources. All of the folders must be on place without exception. It is important to point out that the packages files for L1FE and CTOC must in the folder 'src' directly.

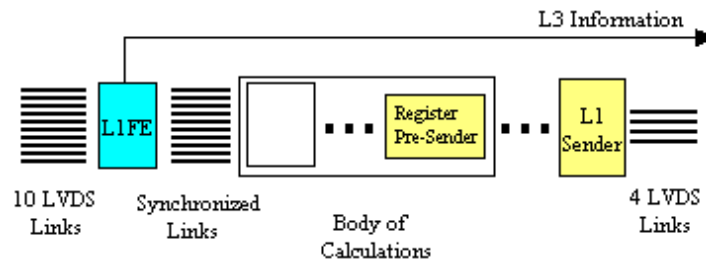


**Figure 8** Folders arrangement

In order to do coding and the corresponding simulations, this structure requires to add some new folders, which will contain the VHDL Test Bench [\[14\]](#) and the Test Vectors Data.

## 5 L1 CTOC VHDL Components

The below figure shows the general idea of the coding, so it is possible to differentiate three clear stages arranged according to their functionality. The module at the input called Level 1 Front End (L1FE) is basically a sophisticated receiver; the body of calculations contains the main entities for calculations, and a 'Register Pre-sender', which organizes the data output bus; the last step is the 'L1 Sender', which makes the Vertical Parity calculation and organizes the output according to the protocols.



**Figure 9** Basic structures

### 5.1 L1FE

This module [6] has three specific tasks

- Synchronization and de-skew
- Level 3 Sender
- End of Record marker

It takes 10 LVDS inputs; this value is a constant, which can be changed according to the user requirements, and its instantiation must be made in the top-level entity, in this case 'L1CTOC.vhd'. The L1FE also contains a powerful Level 3 Sender that can fix the length and the depth into the FIFO (First Input – First Output) arrangement of data for specific study objective. In the L1CTOC case, these values are set by default, so the length to the bus is fixed as the copy of 10 links coming from DFEA, plus header, trailer and padding frames. Also, the depth for the FIFO is fixed to three events before 'L1 Accept' signal appears.

### 5.2 Body of calculations

The requirements about speed have been optimised out into a pipelined structure, which offers a reusability feature. The reusability consists of several entities instantiated with a single 'generic' module, so it helps the compiler to choose a better routing way and as a consequence the implementation is improved. This feature has been tested looking to the speed obtained from both sides; also, the hierarchy inference is easier for the compiler. The basic modules for this kind of generic description are 'pipeline\_BiggestSumPt' and 'pipeline\_Add'.

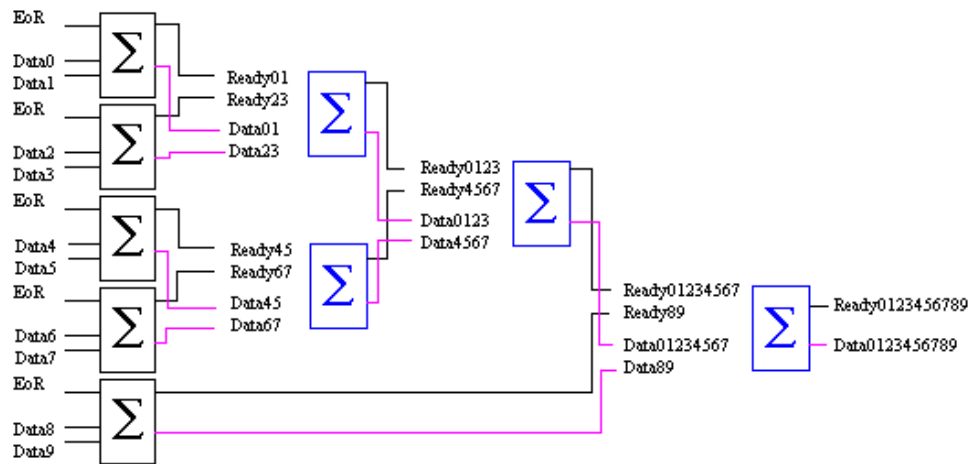
#### 5.2.1 L1CTOC.vhd

This module is the top level. It contains the instantiations for all of the main modules, and their connections to each other, using signals.

### 5.2.2 Pipelined\_Add.vhd

This module uses a generic description for the wide of the input and output buses. It proportions more flexibility and usability. This module use the 'Basic module' showed in numeral [3.1.2](#).

This module uses the configuration showed in the below figure. The signal of synchronization is called 'EoR' and is generated for L1FE. It lets L1CTOC knowing when the output is ready to go out from the memory. It implies that this answer must be synchronized and de-skewed at the time that the 'Body of Calculations' obtains the *ready* signal or 'EoR'. This signal is propagated for is 'Basic module', so the following module will know when the output coming from the previous one is ready to be processed.



**Figure 10** Pipelined architecture.

This module presents a coding structure according in which the 'width' if the input and output bus can be easily changed.

```

component pipeline
  generic (
    width : integer := 10 -- width of the input array
  );
  port (
    dataIn1 : in STD_LOGIC_VECTOR(width downto 1);
    dataIn0 : in STD_LOGIC_VECTOR(width downto 1);
    clock   : in STD_LOGIC;
    ready0  : in STD_LOGIC;
    ready1  : in STD_LOGIC;
    readyOut: out STD_LOGIC;
    dataOut : out STD_LOGIC_VECTOR(width downto 1)
  );
end component;

```

**Figure 11** VHDL - pipeline module

### 5.2.3 Pipeline\_BiggetSumPt.vhd

This module basically has the same concept of the previous module. However, this module does not make additions only performs a choosing process, selecting which is the biggest received value at its input.

```

component biggest_sorter
  generic (
    width1 : integer := 10; -- width of the input array #1
    width2 : integer := 10; -- width of the input array #2
  );
  port (
    RAS0      : in STD_LOGIC_VECTOR(width1 downto 1);
    RAS1      : in STD_LOGIC_VECTOR(width1 downto 1);
    SumPt0    : in STD_LOGIC_VECTOR(width2 downto 1);
    SumPt1    : in STD_LOGIC_VECTOR(width2 downto 1);
    ready0    : in STD_LOGIC;
    ready1    : in STD_LOGIC;
    clock     : in STD_LOGIC;
    RAS       : out STD_LOGIC_VECTOR(width1 downto 1);
    SumPt     : out STD_LOGIC_VECTOR(width2 downto 1);
    readyOut  : out STD_LOGIC
  );
end component;

```

Figure 12 VHDL - pipeline module BiggestSumPt

### 5.2.4 L1CTOC\_EoR.vhd

This module is in charged of stretching the 'EoR' signal generated by 'L1FE'. L1CTOC and its pipelined conception require having a signal a least 6 ticks long while L1FE produces a signal one tick long.

### 5.2.5 Or\_empty.vhd

The outputs of this entity avoid to have undefined wires going as inputs for CTTT. This happens because the compiler optimises out the wires that do not carry information. It takes an input and makes an 'OR' of it, then it is propagated to the Level 1 output.

### 5.2.6 Pt Sum.vhd

This module uses '[pipelined\\_add.vhd](#)' in order to do the additions of its inputs. The value of 'width' for this module is equals to '7', although the input of this module is only 6 bits wide. The new bit is required for taking the 'carry' output of the additions.

Its inputs are coming from each trigger sector in the frame number '6', giving the value for the [Sum Absolute Pt](#) in a six bit wide word. Its output is reported to CTTT in frame number '4'.

An overflow output is generated when the addition in whichever step of the pipeline is more than '63' (decimal value), then the output of this module would be "111" (binary value).

### 5.2.7 Tracks\_sum.vhd

This module takes the information coming from is [Pt Bin](#) input and makes its corresponding output.

This module uses '[pipelined\\_add.vhd](#)' with a width of '4'. Also, it uses VHDL 'generate' instantiations, which, in this case, let the compiler generate multiples structures with the same architecture.

According to the previous assumption, this module repeats 24 times the same structure for this addition.

An overflow output is put when the any of the addition values, in whichever step of the addition process is greater than '7'. If this happens, the addition output becomes "111" (binary value).

### 5.2.8 Occ\_Lev.vhd

The occupancy module takes the information about the number of hits obtained in each trigger sector of that specific octant. This information is added up and truncated in order to generate the trigger terms that are going to CTTT.

This module uses '[pipelined\\_add.vhd](#)' in order to do the additions, and an overflow output ("1111" – 4 bits wide) is generated when any of the additions is greater than '1024' (decimal value) hits per octant. Otherwise this module takes the value of the addition and divide it by '64' (decimal value).

Its 8 bits input information is coming in frame number '6', bits 15 to 8, and its 4 bits output is coming out in frame number '2', bits 5 to 2.

### 5.2.9 PS.vhd

This module makes the addition of the Pre-Shower cluster found. Its description uses the entity '[pipelined\\_add.vhd](#)' in order to do such arithmetic operations. If this value is bigger than '8', an overflow output is generated, so the value "11" (binary value) would be showed.

Its 2 bits wide inputs are coming in frame number '4', bits 0 and 1, and its 3 bit wide output is placed in frame 3, bits 2 to 0.

### 5.2.10 Biggest\_SumPt.vhd

This entity differs from the previous 'pipeline basic module' users, for it requires a new input and output and a different internal process, which does not require additions. Instead of it, that output must be come from the selection of the biggest value with its corresponding Relative Address Sector value. This module is based in '[pipeline\\_biggestSumPt.vhd](#)'.

The inputs of [Sum Absolute Pt](#) are coming in frame number 5, bits 23 to 18, and the inputs for the 'Relative Address Sector' and the 'Octant ID' are coming in the Header (frame 1), bits 21 to 19 and 11 to 8 respectively. Its output corresponds to the Biggest Sum Pt location, which is placed in frame number '6', bits 20 to 17.

Also, a signal of OCT ID error is generated when the embedded value does not match the DFEA output. This flag is placed in frame '6', bit 16.

### 5.2.11 Iso\_Tracks.vhd

This module chooses the isolated tracks detected. This selection is made taking two neighbours, which are arranged as follows.

**Table 12** Isolated tracks

L1CTOC Output	Isolated information source (Frame 6, bits 7 to 0)
Frame 5, bits 23 to 16	Link 0 and Link 1
Frame 5, bits 15 to 08	Link 2 and Link 3
Frame 5, bits 07 to 00	Link 4 and Link 5
Frame 6, bits 15 to 08	Link 6 and Link 7
Frame 6, bits 07 to 00	Link 8 and Link 9

### 5.2.12 Gl.vhd

This module signs if the information found in an 'isolated track' corresponds to a given Occupancy value. The information that it requires are the bits 7 and 0 coming in the frame '6', which corresponds to the isolated track information.

### 5.2.13 Header\_MV.vhd

This module was conceived in such a way that it would use Majority Vote in order to generate its output. This feature must be implemented in a near future. By now, this value is obtained directly from Link number '2', which carries the Global Clock.

Its inputs are coming in frame number '1' (Header). They are 'Pass Mark' information (bits 23 to 22), 'Serial Command Link' information ('CFT Reset' in bit 18 and 'First Crossing' in bit 17), and 'Data Type' (bits 7 to 3).

The output for the SCL information is placed in frame '2', bits 2 to 0.

### 5.2.14 Revision\_reg.vhd

This module has been provided by Jamieson Olsen. Its main function is providing the firmware version number for each firmware design.

Also, the latest versions of L1CTOC uses it for choosing the Test Vectors implemented into the 'regPresender.vhd'

### 5.2.15 RegPresender.vhd

This module organizes the information coming from all of the modules in the 'Body of calculations' and generates a single bus of 144 bits wide.

Also, it is the responsible for choosing the mode of operation using the information coming from the 'Firmware Revision Register'. Those values are showed in the below table.

**Table 13** L1CTOC Modes

Firmware Revision Value (Hexadecimal value)	Level 1 output	
00	Normal mode	
90	Test Mode	Valid Zeros
A0		Event A+B
B0		Busy Event

## 5.3 L1 Sender

This module generates the vertical parity and formats the information according to the protocols in order to send it to the next stage CTTT.

It is a simple Finite State Machine, which does not use reset, so it has "software reset".

## 6 Configuration Parameters

The main values to be affected are in 'L1FE\_pack', which correspond to the Number of links coming from DFEA, Number of links to be sent to Level 3, Number of Frames of each input, and the depth into the FIFO for Level 3 effects. However, this changes must be do in the top-level entity instantiation of the module.

At the L1CTOC side, there is a value that must be configured in order to get eight different flavours of L1CTOC. This value is in 'CTOC\_pack', and it can go from "000" to "111" (binary notation).

## 7 Scripts

These tool, wrote by Levan Babukhadia, let more flexibility and precision in order to choose the synthesis and implementation requirements.

These scripts depends of the location of the files, so the user most change the path in which the files are for 'implement.bat' and 'synthesis.bat'. Also the same path must be placed in the file 'L1CTOC.fes', which calls the files requires for synthesis and describes some of the requirements in such process.

## 8 Timing Simulations

### 8.1 Latency

The total latency is estimated to be 2.5 crossings. If it takes a global clock with a period equals to 20ns, since the frequency is close to 53 MHz (18.8ns), this value is close to 350ns.

$$Latency = (Gclk\_period) * (frames) * (2.5)$$

$$Latency = 20ns * 7 * 2.5$$

$$Latency = 350ns$$

However, the real implementation showed a value equals to 3.5 crossings, and then the latency becomes 490 ns.

### 8.2 Body of calculations latency

This calculation does not take account of L1FE.

## 9 1553 interaction

The communication is made by 1553 protocol of communication, which talks directly to the controller board (DFEC) [\[12\]](#) in DFE crate. Most of the information is propagated using a interface software denominated DFEWare [\[13\]](#), which offers interactivity with the boards placed in such crate for d0online users.

The designs are downloaded using this interface. They must be in a proper format, so it must be generated by Xilinx PROM file formatter, which generated a \*.HEX file.

Once the process makes the file, it must be agree with the following naming conventions.

**CTOC\_PW03\_2\_2\_D0\_v000.004**

Board name                      Crate                      Slot                      Device                      Version

**Figure 13** DFEWare convention

The above example is placing a file with Version "004", into device D0, slot number '2', crate number 'PW32', for CTOC board.

Device number '0' and Device number '1' are filled with blank designs, U3 and U5 respectively, while Device number '2', corresponding to U1, is the place in which the design is going to be set up.

## 10 Implementation issues

Because each Octant has its own ID number, they must be compiled in separated process. This implies that the speed for each L1CTOC, although they have the same basic code, change for the different values of the ID number. The worst cases are when ID equals to "000" and "010".

The average speed for latest designs, until CTOC version 61, is 60 MHz, implementing the L1CTOC design for a FPGA Xilinx Virtex 600BG560-5.

The number of RAM memories used is 20 of 24. Global buffer clocks are fully used, together with their corresponding global net.

Some outputs can be looked in the Front End Board Panel thanks to the routing through the Mother Board, using U49 CPLD. It is important to point that any signal put into this bus is stretched, so it does not require a previous process in order to be looked by the human eye using LEDs.

## 11 Outlook

The L1CTOC has been successfully tested for Fake tracks, A and A+B events. Also, its L3 Sender has been read over FIC [10] into L2 crate [11].

Some functions require be studied for implementation, and depth studies at the online and offline level are strongly recommended.

Also, the featurization of 'L1 Accept' signals arriving to design and different values of Firmware Revision Register for test mode must be tested.

## 12 Glossary

**Table 14** Generic Board Names [\[15\]](#)

Term	Meaning
DFEA	Digital Front End – Analog
SWDB	Single Wide Double Board
MB	Motherboard
CTOC	Central Tracker Octant Card
DWDB	Double Wide Daughter Board
AFE	Analog Front End
FIC	Fiber Input Converter Card
SCL	Serial Command Link
CTTT	Central Tracking Trigger System
DFEC	Digital Front End Controller

**Table 15** Communications Protocols

Term	Meaning
LVDS	Low Voltage Differential Signalling
G-Link	HP G-link - Fiber optic based fast link
1553	Mil-Std-1553
FSCL	Fast Serial Copper Link

**Table 16** Others

Term	Meaning
CPS	Central Pre-Shower
CFT	Central Fiber Tracker
FPGA	Field Programmable Gate Array
CPLD	Complex Programmable Logic Device
DFEWare	Software for communication via 1553

## 13 References

- [1] CTTT Group, "CTT technical design report," D0 Note 3551, Jan. 1999.
- [2] Xilinx, Virtex Data Sheet, Feb. 2002.
- [3] J. Olsen, " CFT/CPS sector and singlet naming convention. LVDS bus colors and motherboard input bus structure defined.," Dec. 2001.
- [4] Xilinx, "Xilinx Fine-Pitch BGA and CSP Packages," Mar. 2000.
- [5] L. Babukhadia, " DØ Track and Preshower Trigger Level 1 Trigger Terms and Data Transfer Protocols V07-00", Apr. 29, 2002 "
- [6] L. Babukhadia, M. Martin, and S. Desai, "L1FE design (VHDL design)," D0 Note 3881, Jul. 2001.
- [7] J. Olsen, "Proposed method for de-skewing LVDS datastreams in a Virtex FPGA," Jan. 2001.
- [8] S. Rajan, Essential VHDL - RTL Synthesis Done Right,, pp. 21 - 50, 1998.
- [9] CTT Group "CTT Technical Design Report, Version 5", D0 Note 3551, Oct. 2002.
- [10] F. Borcharding, M. Johnson, M. Martin, "Links from CFT (CPS) concentrators to L2pp", May. 1998.
- [11] "FIC Card", <http://www-clued0.fnal.gov/L2/fic/>
- [12] J. Olsen, "Digital Front End Controller Card",  
<http://d0server1.fnal.gov/users/jamieson/www/projects/dfec/index.html>
- [13] D. Evans, Y. Maravin, [http://www-clued0.fnal.gov/dfc\\_ware/](http://www-clued0.fnal.gov/dfc_ware/)
- [14] L. Babukhadia, "General Purpose VHDL Test Bench", D0 Note 3871, May. 31, 2001.
- [15] CTT Group, "Glossary of Terms and Acronyms"  
<http://d0server1.fnal.gov/Projects/TriggerElectronics/WebDocs/Glossary/glossary.htm>
- [16]

## Figures

Figure 1 CTOC Board Disposition.....	2
Figure 2 CFT Octant distribution .....	2
Figure 3 Level 1. Input Protocol. DFEA - L1CTOC .....	3
Figure 4 Level 2. Input Protocols. DFEA - CTOC.....	3
Figure 5 L1 Output. CTOC-CTTT.....	3
Figure 6 Finite State Machine .....	4
Figure 7 Basic Modules .....	5
Figure 8 Folders arrangement .....	11
Figure 9 Basic structures .....	12
Figure 10 Pipelined architecture.....	13
Figure 11 VHDL - pipeline module.....	13
Figure 12 VHDL - pipeline module BiggestSumPt.....	14
Figure 13 DFEWare convention .....	18

## Tables

Table 1 Pass and Mark.....	6
Table 2 Serial Command Link .....	6
Table 3 Occupancy Level .....	7
Table 4 Data Type .....	7
Table 5 Level 1 / Level 2.....	8
Table 6 Bits - Isolated Tracks .....	8
Table 7 Isolated Track Frame .....	8
Table 8 CPS cluster association .....	9
Table 9 Pt Bin.....	9
Table 10 Extended Pt.....	10
Table 11 CPS association .....	10
Table 12 Isolated tracks.....	16
Table 13 L1CTOC Modes.....	17
Table 14 Generic Board Names [15] .....	20
Table 15 Communications Protocols .....	20
Table 16 Others .....	20